BOOK ONE · A FIRST TEST CASE IN SOFTWARE

Designing Coherence

How to operate when
AI does the execution

RUTH AMICHAY

First Edition · 2026

PRE-LAUNCH EDITION

A note before you begin

This is a pre-launch edition of *Designing Coherence*, sent to a small group of one hundred readers ahead of public release.

You are reading it because your perspective matters to me. I would love to hear your impressions — what landed, what didn't, what felt true, what felt off, what you'd want to see more of, and anything that made you stop and think.

Comments, questions, disagreements, and quiet observations are all welcome. Even a single sentence helps shape what comes next.

Thank you for reading it early.

Ruth Amichay
ruthamichay.com

Designing Coherence

How to operate when AI does the execution

Book One · A First Test Case in Software

Ruth Amichay

First Edition · 2026

Book One in the *Designing Coherence* series. A first test case, focused on software and SaaS companies.

Contents

Prologue

Something is off. It is hard to name, but easy to feel. The company is moving — faster than ever. AI is working. Features ship in weeks instead of quarters. Prototypes appear overnight. There is no shortage of momentum.

And yet the machine does not feel right. More is happening, but it is not clear that more is adding up to better. Leadership meetings cover more topics but resolve fewer questions. Decisions are made, but it is hard to tell whether they hold through execution.

Some of it is about people. Roles that were essential two years ago feel less clear now. It is harder to say who is truly needed, what their contribution should look like, and where human judgment ends and AI execution begins. Nobody wants to ask the question out loud — but it is in the room.

Some of it is about direction. Teams have more options than ever, and the temptation to try them all is real. When building is cheap, everything looks worth starting. Direction softens. Focus fragments — not because anyone decided to lose focus, but because trying anything has become affordable.

Some of it is about control. The indicators that used to signal progress — velocity, time to market, story points completed — were designed to track execution. When AI handles the execution, those numbers still move, but they stop tracking what actually matters. The old scoreboard is running. It is just measuring the wrong game.

And underneath all of it — the potential is enormous. Everyone can feel it. AI can change the economics of every function in the company — product, engineering, sales, marketing, customer success. The wiring, however, was not built for this voltage.

The operating model — the way decisions get made, the way teams stay connected, the way ownership works — was designed for a world where execution was slow, expensive, and siloed by necessity. That model served us well when humans did the execution. It does not naturally fit a world where execution is done by AI.

> The question is not whether to adopt AI. The question is how to operate a company once AI has changed what it means to know, own, and decide.

PART ONE

The Shift

What held companies together,
what changed, and what must replace it

THE THREE FORCES

What Kept It All Working

For decades, every new idea required a substantial investment — real people, real budget, real months of work. Before anything could start, someone had to ask whether it was worth the cost, the effort, and the time.

Three forces shaped this reality. Nobody planned them. Nobody named them. They were side effects of how execution used to work — cost, time, and the way knowledge was held in professional silos. Let's look at each one.

Cost gave direction.

Building a feature required a team, several sprints, and a budget that was impossible to ignore. That cost acted as a natural filter. Not everything that seemed like a good idea could actually start. Someone had to choose — and that choice created focus. Only the ideas that survived serious evaluation made it to execution. The weight of potential failure made people plan carefully, challenge each other honestly, and think twice before committing.

Time gave alignment.

From idea to customer, months passed. Each step involved different people, and each handoff created a natural pause — a moment to check whether things still made sense. These pauses were not designed as governance.

They were a side effect of how long things took. But they gave people time to align, catch mistakes, and ensure that what was being built matched what was intended.

Silos gave accountability.

No single person could hold all the signals a company produced. The volume of information — customer feedback, technical constraints, market signals, operational data — was simply too large for any one mind. So organizations divided the world: each function held their slice, became expert in it, and made decisions from within it. Nobody knew everything — and that was not a failure. It was the only rational response to cognitive overload.

> Siloed knowledge made accountability easier to assign. You owned your domain because only you could see it. Accountability came with the territory.

A MARGIN, NOT A GUARANTEE

What These Forces Really Did

None of these three forces was a strategy. None was named in any operating manual. They were the side effects of how slow, expensive, and bounded execution used to be. And yet, taken together, they did the work that an operating system would otherwise have to do.

Products that should have shipped misaligned often got caught in time. Teams that should have fragmented were forced back together at the next handover. Decisions that would not have held got revisited at the next budget review. The three forces did not produce well-run companies. They produced companies where being badly run was less fatal than it should have been.

This margin held because execution was the constraint, and the three forces kept it in check. AI weakens all three at once. With each one that weakens, a new risk appears.

THE THREE FORCES, IN SUMMARY

Cost	Time	Silos
gave direction	gave alignment	gave accountability

THE IMPACT OF AI

What Changed

AI did three real things, and each one is a gift before it is a problem. It made execution close to free. It made it close to instant. And it gave everyone access to context that used to be locked inside one function. Each of these opens up something genuinely new. Each one also weakens one of the three forces that used to hold companies together.

Cost approaches zero, and the company can explore more directions than ever before. Ideas that would have died at a budget meeting can be tried in an afternoon. The space of possibilities widens. But the same drop in cost means cost no longer filters. Everything looks worth starting, and the company drifts toward doing all of it. When trying is cheap, the decision itself becomes optional. Teams hedge: let's try both, prototype all three, see what the market says. The cost of not committing collapses along with the cost of trying. Strategy does not just diverge into too many directions; it never really chose any.

Time collapses, and every function can work on the same initiative at the same time, with a real picture of the final shape early in the process. The team sees what is being built before it is finished — together, in real time. But the same simultaneity removes the natural pauses where the streams used to check each other. Each function moves fast in its own direction. The work begins to fragment.

And context is no longer siloed. A salesperson runs a market analysis. An engineer drafts a spec. A marketer prototypes a feature. The boundaries between professions blur, and everyone has the full picture they need to act. The risk is the other side of that gift: when anyone can do anything, who is accountable for what?

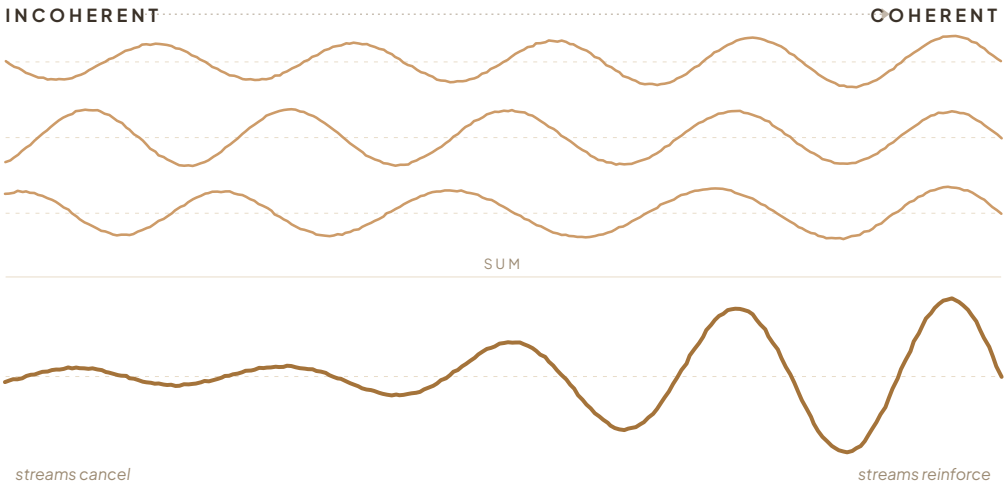
None of these risks is genuinely new. Companies have always struggled with strategy divergence, fragmented execution, and blurred ownership. What is new is that the old margin for error is gone.

What the forces gave us "for free" now needs to be deliberately designed.

Designing how all the parts or ideas fit together well so that they form a united whole is ***designing coherence***.

Coherence is when every part of the company is building one thing — same product, same promise, same direction — at any given moment. It is a property of what is happening right now, across every stream of work. Either the streams add up to one thing, or they do not.

The word is borrowed from physics on purpose. Coherent waves reinforce each other. Incoherent waves cancel each other out. A coherent company is one where the parallel streams of work add up. An incoherent one is where they interfere — even when each stream, on its own, is doing good work.



TWO SIDES OF THE SAME CHANGE

What We Gained, What We Risk

	WHAT WE GAINED	WHAT WE RISK
THEN **Cost gave direction.**	Cheap to try means we can **explore more directions** than ever before.	Nothing filters what to pursue. **Strategy diverges.**
THEN **Time gave alignment.**	Simultaneous work means **a real view of the final shape** early in the process.	No natural pauses to keep streams in sync. **Execution fragments.**
THEN **Silos gave accountability.**	Anyone can act on any kind of work. **Full context is accessible.**	Boundaries between professions blur. **Ownership blurs.**

THE QUESTION

How do we keep what AI gave us — the breadth, the speed, the accessible context — and design the coherence the old margins used to buy us time to figure out?

The next three parts answer that question, one trade-off at a time.

PART TWO

Designing Direction

Choosing which directions earn
the company's attention

WHAT THIS PART SOLVES

Holding Direction Without Killing Diversity

Of the three risks the shift creates, the one this part addresses is **strategy divergence**. Cost no longer filters. Anything looks worth starting. The company drifts toward doing all of it and not completing anything.

The temptation is to handle this with more process — more reviews, more sign-offs, more permission. That would solve divergence by killing what AI gave us. The whole point of cheaper exploration is that the company can try more directions, follow unexpected paths, and see what works. That breadth of options — the diversity of paths the company can now explore — is the gain. We do not want to block it. We want to keep it, and still hold direction.

To do that, we have to look more carefully at how producing something actually works. Many parallel streams of work are moving at once — across functions, across initiatives, across stages. Most of the time they run in parallel. But there are a few — only a few — natural moments where they have to come together: when a decision has to be taken, the streams converge, the right people look at the same picture, and a call is made. These are **convergence points**.

Convergence points are not everywhere. They are deliberately sparse — between them, the work is free to evolve, explore, and change shape. They sit only at the moments where the streams have to meet. There are three of them.

THE FLOW

Define · Produce · Operate

Producing anything goes through the same flow. Someone defines what to do. Someone produces it. Someone uses it. Same person or different people — the flow does not change. It does not change with the size of the company, the type of product, or the era of execution.

In an operation, those three stages translate into specific work.

Define

What we want to do. Why we want to do it. Who it is for. When it needs to happen. The stage where intent becomes a plan.

Produce

Code. Pricing. Pitch. Demos. Manuals. Playbooks. Everything the company needs to put the plan in front of a customer.

Operate

Sell. Onboard. Maintain. Measure. The stage where what was produced meets the people it was produced for.

WHAT CHANGED

AI at Every Level

Until recently, each stage had its own people, its own constraints, and its own ceiling. The number of directions a team could explore, the number of versions it could build, the number of signals it could act on — all of these had limits. Cost was the limit. AI removes it. At every stage, what used to be one or two options is now ten. What used to be a draft is now a dozen drafts.

IN DEFINE

It proposes directions. It analyses internal and external data. It drafts requirements. Every meeting can begin with twenty viable plans instead of two.

IN PRODUCE

It proposes architecture. It writes code. It drafts marketing campaigns and sales decks. Three versions of the same campaign, five flavours of the same feature — all at once.

IN OPERATE

It onboards customers. It monitors anomalies. It compares results against success criteria. It surfaces signals from every channel at once — usage, support, sales, sentiment.

With cost no longer filtering, selection has to happen on purpose — at three deliberate moments where the work either crosses from one stage to the next, or stops.

The Three Convergence Points

Three convergence points. Each answers one question. What starts. What ships. What survives. At each one, parallel streams meet, the right people look at the same picture together, and a call gets made. These are not approvals from above. They are moments of shared judgment.

Each convergence point shares the same shape. The work either passes or it does not. The criteria are written before the room meets — at the previous convergence point, or at the start of the work — so the decision is enforced rather than negotiated. The room does not invent the standard. It applies it.

This is what makes the moments holdable. If criteria were debated in the room, the strongest voice or the most senior title would prevail, and the convergence point would become an approval. Writing them in advance puts authority in the criteria, not the people. The room becomes a place of reading, not arguing.

1 · Selection Point

Should this start?

A working session where the people who will deliver the work commit to it in front of each other. It is not approval from above. If anyone in the room cannot say *"I am confident I can deliver my part,"* the work does not start. It is sent back, reshaped, or killed.

WHAT GETS NAMED AT SELECTION

- The customer signal that justifies the work — what we heard, from whom, how often
- The success threshold — the specific conditions that, six months from now, will tell us this was worth doing
- The kill threshold — the specific conditions below which the feature ends without further debate. Written now, while no one is yet attached.
- What we are not doing instead — the explicit trade-off, named
- The owners — which functions on the team are accountable, and what each one is committing to
- The Validation date — set now, not later. The day we will look back and decide whether this lives

2 · Shipment Point

Is this ready?

No single function can release alone. Readiness is a shared state, not a function-by-function checklist. Engineering ready does not equal shippable. Sales ready does not equal shippable. Shippable means every function holding its part of the launch — at the same time, against the same definition.

If anything is missing, the work waits. Shipping incoherent is worse than shipping late. A launch where Marketing told one story, Sales told another, and Support was never briefed is not a launch — it is the company telling its customers, in three voices, that it does not know what it is doing. The Shipment Point exists to catch this before it reaches the customer.

WHAT GETS CONFIRMED AT SHIPMENT

- Technical readiness — does it work, and have we tested what we said we would test at Selection
- Operational readiness — can support handle the inbound, can the team operate it on day one, are runbooks current
- Commercial readiness — is sales briefed, is pricing live, is the offer real and consistent across channels
- Customer readiness — do the people we promised this to know it is coming, in the way we promised it
- Coherence — does what we are about to ship still match what we committed to at Selection — or has it drifted

3 · Validation Point

Does this live, or does it die?

Most companies revisit shipped work — in QBRs, portfolio reviews, post-launch retrospectives. Very few have a kill function. Things accumulate, attention thins, and features that should have ended quietly continue. The Validation Point is the scheduled moment where the criteria written at Selection — including the kill threshold — are surfaced unedited and compared to what actually happened. Below the kill threshold, the decision was already made. Above it, the team makes the call in the open: keep, kill, or evolve.

WHAT GETS COMPARED AT VALIDATION

- What we promised at Selection vs. what actually happened — adoption, revenue, the success criteria we named
- What we learned that we did not know at Selection
- The cost of keeping it alive — engineering, support, sales attention, cognitive load
- The opportunity cost — what could we do instead with the same attention

> Validation does not ask the team for courage. It asks the team to honor the criteria they already wrote.

DECISION POINTS · THE KILL THRESHOLD

Designing the kill criteria

Killing a working feature is the hardest organizational act in software. The Validation checkpoint, by itself, does not solve this.

The hard case is not the failed feature. The hard case is the feature that works — has real users, generates real revenue — and still should not exist. Because the metric was the wrong metric. Because the maintenance cost exceeds the value it returns. Because the attention it consumes is attention not spent on what would actually move the company.

By the time Validation arrives, the team is attached to what they built. Asking them to write honest kill criteria in that moment is asking too much. So the kill criteria get written earlier, at Selection, when nothing is yet at stake.

Selection now names three zones defined by two thresholds, not one — a success threshold above which the feature is confirmed, a kill threshold below which it ends, and a continuation zone in between. The narrower that middle zone, the sharper the team's direction. Narrow zones require courage; the first cycles are the hardest. The zone tightens as the team learns to trust its own criteria. The kill threshold does most of the work, before any meeting is convened.

> Courage at Selection is cheaper than courage at Validation. The kill threshold is where the courage lives.

PART TWO · CLOSING

How Direction Is Held

Strategy divergence happens when nothing chooses. The convergence points are how the company chooses without slowing the work down between them.

Direction is no longer maintained by cost or by approval cycles. It is maintained by three moments where the company looks honestly at what it has, and decides whether to start, ship, or keep.

Those are not stage-gates. Stage-gates were designed to control risk on slow, sequential projects by adding sign-offs between phases. Convergence points do the opposite work: they protect the speed and parallelism between them.

This is the first kind of coherence the framework rebuilds: **directional coherence** — every stream of work pointing at the same direction the company has chosen to pursue.

> Diversity is preserved between those moments. Direction is held at them.

PART THREE

Designing Alignment

Rebuilding how work moves:
holding parallel work together

WHAT THIS PART SOLVES

Holding Parallel Work Together

Part Two was about *what* gets built — how the company chooses its direction at three moments and protects diversity between them. This part is about something different: how the work stays one thing *while* it is being built. Of the three risks the shift creates, the one this part addresses is **fragmented execution**. With AI, every function can work on the same initiative at the same time. That parallel work is the gain. The risk is that each stream is correct on its own and none of them is in conversation with the others.

The obvious solution is to slow things back down — re-introduce sequence, add handoffs, make each function wait for the previous one. That would solve fragmentation by giving up what AI gave us. The whole point of simultaneous work is that the company learns faster, sees the final shape earlier, and can react in real time. We do not want to break the parallelism. We want to keep it, and still hold the work together.

To do that, the company needs an operational structure built for parallel work — a way of working that keeps the streams aligned without blocking them. The rest of this part is what that structure looks like.

> We do not want fewer streams. We want them to stay one company.

Execution Is AI

The stages of work have not changed. Define, Produce, Operate. An initiative is still defined before it is produced, and produced before it is operated. That sequence is true in any era. What has changed is what happens *inside* each stage.

Functions inside a stage used to work in sequence because each one needed the output of the previous one before they could move. Marketing could not write copy before the feature existed. Sales could not pitch before the product was real enough to demo. Working in parallel was wasteful — because every change upstream forced expensive re-work downstream. The cost of staying current was higher than the cost of waiting.

AI inverts that math. When the cost of change is negligible — when a tightened scope or a refined pitch can regenerate every dependent artifact at almost no cost — every function gets to enrich every other function continuously. The salesperson notices something while pitching. The marketer sees a sharper way to position the same feature. Each is a small gain. Together, in real time, they make the work much better than any one function could make it alone. The cost of re-work is now lower than the value of mutual enrichment — and so the work begins to evolve together rather than wait for each other.

> In parallel, every function enriches every other, continuously.

THE EVOLUTION

Three Eras of Execution

To see why simultaneous execution is different, it helps to look at what came before. Every era of software has been defined by how work moves. Each era found its own way to keep the pieces aligned. None of those ways works now.

GENERATION 1

Waterfall

Sequential

One phase finishes before the next begins. Requirements, then design, then build, then test. Slow — but coherence was built in. Each handover forced alignment.

Held by sequence

GENERATION 2

Agile

Iterative

Short cycles. Build a little, learn a little, adjust. Faster, more adaptive — but coherence depended on the team staying close and communicating constantly.

Held by communication

GENERATION 3

AI era

Simultaneous

Everything happens at once. AI drafts code while another agent tests it while a third writes documentation. No sequence. No natural pause. No built-in alignment.

Held by design

Waterfall gave coherence through sequence. Agile gave coherence through communication. Neither works when execution is simultaneous. Something else has to do that work — by design.

THE DISCIPLINE

What Holds Simultaneous Work Together

Holding many parallel streams together requires a common source of truth — one place where the truth of the work lives. That source is kept current as the work moves, and surfaces changes the moment they happen.

One source of truth

No private copies. No parallel documents. Whatever needs to be true about the work is in one place — and only there. Anyone who needs to act on the work looks at the same picture.

Continuously updated

What is shared has to reflect what is currently true. Not what was true last week. As the work moves, the picture moves with it — without anyone having to remember to update it.

Changes well managed

When something shifts, the streams that depend on what shifted are told. Other streams keep moving. The implications of every change are visible to the work that needs to see them.

THE SHARED SOURCE · 1 OF 2

The Book of Record

When work happens in many parallel streams, the company needs one place where the truth of the work lives. Without it, every function holds its own version of what is true, and the versions drift. With it, anyone who needs to act on the work — at any moment, in any function — opens one record and sees where things stand.

A **book of record**. One per initiative. One source of truth, continuously updated by the people doing the work, holding only what is currently relevant to the stage the initiative is in.

It is not a log. It is not an archive. As the initiative moves from one stage to the next, what is no longer relevant gets pruned. What survives is the decision committed at the convergence point and the truths that still matter. A working document, not a historical one.

This kind of artefact was not practical before AI. Keeping a single document this rich and this current would have cost more than it returned. So companies settled for fragments: a PRD here, tickets there, a wiki page that went stale in a week. AI changes the economics. It can watch every channel where information about the initiative lives — the issue tracker, the support queue, the sales calls — and keep the record current. The work that used to live in twenty places now lives in one.

> One source of truth, kept current, holding only what matters now.

THE SHARED SOURCE · AN HONEST ASIDE

Science fiction, or operating artifact?

The Book of Record is the part of this framework most dependent on AI capability still maturing.

Anyone who has tried to point an LLM at Slack, Jira, Linear, CRM data, support tickets, and customer call transcripts knows how aspirational that still is. AI summarization drifts. It double-counts. It hallucinates entities. It misses nuance. It silently goes stale when the upstream channels change.

So the question is not whether the Book of Record exists today in finished form. The question is: what does the operating discipline look like when AI gets it 80% right, not 100%?

The answer is a division of labor. AI keeps the currency — what is fresh, what changed, where the signals are now. Humans keep the meaning — what the signals are saying, which ones matter, what to do about them. The Book of Record is not a system that replaces judgment. It is a system that frees judgment from the work of gathering.

> AI is the keeper of currency. Humans are the keepers of meaning.

What the Record Carries in Each Stage

Here is what the record is reaching toward, even where today's tools only get part of the way. Each stage produces a different kind of work. The record carries what matters at that stage, and lets go of what no longer does at the convergence points between them.

In Define *building shared understanding of what to do and why*

- The problem being solved and who it is for
- What success looks like — the criteria by which we will know
- The proposed solution and the alternatives considered
- The constraints, risks, and open questions still being resolved
- The proposed scope, timeline, and owners by function

CLOSES AT THE SELECTION POINT — DOES THIS START?

In Produce *making the thing, with all functions working in parallel*

- The committed scope (a commitment now, not a draft)
- What each function is producing — code, copy, onboarding, support content, pitch
- The dependencies between streams and what each one depends on
- Changes since the Selection Point and which streams each change touches
- Readiness state and the open risks that could delay shipment

CLOSES AT THE SHIPMENT POINT — IS THIS READY?

In Operate *the initiative is live; the question is whether it works*

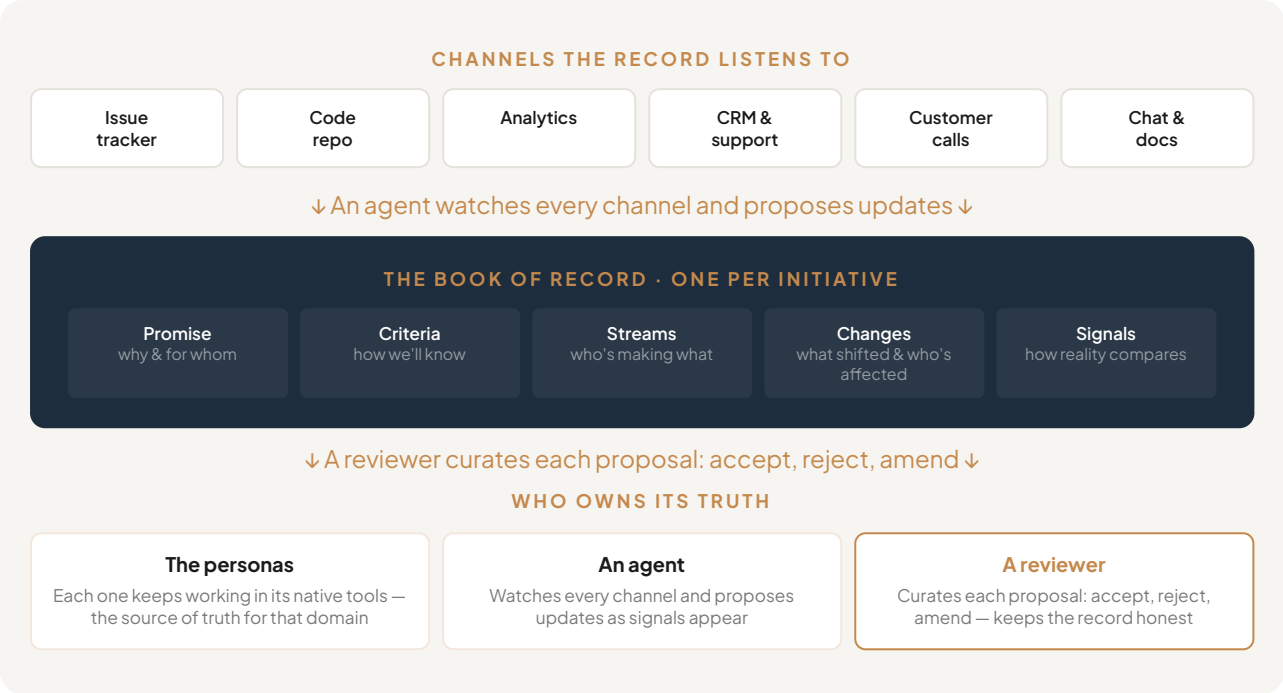
- What was promised at the Selection Point against what is happening
- Adoption, usage, and the customer signals coming in
- Support load, performance, and reliability
- Lessons that have emerged about the customer or the market
- Open issues and the decisions still to make

CLOSES AT THE VALIDATION POINT — DOES THIS LIVE OR DIE?

HOW IT ACTUALLY RUNS

What Maintains the Record

The book of record is a working artefact, not a wiki. Three layers keep it alive: the channels it listens to, the record itself, and the people who own its truth. None of these is new technology. What is new is the discipline of pointing them all at one document.



The record is not a tool — it is a discipline that several tools can carry. A team running on Linear, Notion, and Slack can hold a book of record as well as one running on Jira and Confluence. The same is true of the agent that watches those tools and surfaces signals: it is an execution-layer capability the company builds or buys around its own stack — not part of this book. What matters is the agreement that **this is the document**, that an agent feeds it from every channel, and that someone is accountable for keeping it true.

DURING EXECUTION

When Something Changes

Once work begins, things change. A technical approach turns out to be wrong. A customer signal contradicts an assumption. A market shifts. The team learns something it did not know on the day the spec was written.

In sequential execution, changes were absorbed by the next phase. In iterative execution, they were absorbed by the next sprint. In simultaneous execution, there is no "next" — every stream is already running. A change has to be handled in real time, and the question is always the same: who needs to know?

Two kinds of change

Implementation change

A change in *how* we produce it. The output looks the same, the path to it is different. Affects only the function building it. Other streams continue without interruption.

Specification change

A change in *what* we are producing. The output is no longer what was promised. Every affected function has to re-confirm — through the book of record — that they can still deliver their part.

The book of record makes the difference visible. Every change is logged. Every change is classified. The record tells everyone — without a meeting — whether they need to do anything about it.

> Parallel work survives change because the record tells you exactly who is affected.

PART THREE · CLOSING

How Parallel Work Stays One

Fragmented execution happens when parallel streams stop being in conversation with each other. The book of record is how they stay in conversation without slowing each other down.

When every stream works from the same record, no stream is acting on a private version of the truth. When the record is continuously updated, no stream is acting on a stale picture. When changes are well managed, every stream that depends on what changed knows — and the others are not interrupted. When the record carries only what matters now, no stream is distracted by what no longer matters.

Alignment is no longer maintained by sequence, or by handoffs, or by everyone being in the same room. It is maintained by one source of truth, well operated.

This is the second kind of coherence the framework rebuilds: **executional coherence** — every stream of work acting on the same picture of reality at the same moment.

> The streams stay parallel. The company stays one.

PART FOUR

Designing Accountability

Rebuilding ownership:
personas, and the Band that holds them

WHAT THIS PART SOLVES

Whose Judgment Is It?

Of the three risks the shift creates, the one this part addresses is **blurred ownership**. With AI, anyone can act on any kind of work. The salesperson runs a market analysis. The engineer drafts a spec. The marketer prototypes a feature. Context is accessible to all. The boundaries between professions blur. And when anyone can do anything, who is accountable for what?

The obvious solution is to put the old org chart back together — define silos clearly, mark who owns what, and let the boundaries do the work. That would solve blurred ownership by giving up the breadth AI gave us. The whole point of accessible context is that anyone can act on the right information at the right time. We do not want to put the silos back. We want clear ownership without losing the breadth.

To do that, we have to ask a different question. Not *which department owns this* — but *whose judgment is required for the call to be made*.

> Departments are about execution. Ownership in the AI era is about judgment.

WHERE HUMANS STILL WIN

What AI Does. What Humans Do.

Before deciding how to fix ownership, look at what is actually happening. Across every profession in a software company, work splits into three zones. Human only. Human and AI together. AI alone.

PROFESSION	HUMAN ONLY	HUMAN + AI	AI ALONE
Product	Strategic direction. Roadmap calls. Trade-off judgment.	Spec writing. Customer-need synthesis. Prioritisation framing.	Feature documentation. Change-log drafting. Survey analysis.
Engineering	Architecture decisions. Technical strategy. Long-term trade-offs.	Code review. System design. Debugging hard problems.	Code generation. Testing. Deployment. Infrastructure ops.
Sales	Trust. Negotiation. Discount-vs-strategy calls.	Pitch tailoring. Proposal drafting. Account planning.	Lead scoring. Pipeline forecasting. CRM hygiene.
Marketing	Brand voice. Positioning. Cultural reading.	Campaign concepts. Messaging review. Creative direction.	Content production. Copy variants. Performance tuning.
Customer Success	At-risk customer relationships. Renewal calls. Hard conversations.	Onboarding design. Health scoring review. QBR prep.	Churn prediction. Routine check-ins. Usage reporting.
Operations	Cross-team coherence. Risk-call ownership.	Risk identification. Process design. Escalation management.	Status reporting. Dependency tracking. Operational metrics.
Support	Outage accountability. Escalation ownership.	Root-cause investigation. Customer impact framing.	Ticket triage. Routine responses. Incident detection.

Across every profession, the same kinds of work stay human. The professions look different. The pattern of what stays is the same. That pattern is the seam.

READING THE SEAM

What Stays Human Is Judgment

Look at the human-only column of the matrix again. Across all seven professions, the work that stays has one thing in common. It is judgment.

Strategic direction is a judgment. Architecture is a judgment. The trust between a salesperson and a customer is a judgment. The call about whether something is ready to ship is a judgment. The call about whether what was built actually worked is a judgment. AI can produce the inputs to every one of these. None of them is the AI's call to make.

And the judgments cluster. A product manager deciding what to build, a Customer Success leader deciding when to escalate, a Sales leader deciding when to walk away — different professions, different domains, same kind of work underneath.

Alongside judgment is a second human capacity the framework depends on — the ability to translate a complex situation into language others can act on. AI can summarize. Humans explain. A judgment that cannot be rendered into the room in a way that lets the room respond is, in practice, no judgment at all. Decision and explanation travel together.

> **The seam between humans and AI is not which profession you are in. It is what kind of judgment the work requires.**

NAMING THE KINDS OF JUDGMENT

Personas Are Methods, Not Titles

A **persona** is not a person. It is not a title. It is a kind of judgment.

Judgment is not the same as execution. Execution produces the artifact — writes the code, drafts the copy, runs the analysis, builds the prototype. Judgment decides what should be done, whether what was done is good, and what to do with it. Execution is what AI does best. Judgment is what stays.

EXECUTION	JUDGMENT
Produces the artifact	Decides what to do with it
Solvable by rule	Trade-offs no rule resolves
Owns the output	Owns the outcome
AI does best	Stays human

Six of them in this book — named on the next page.

> A judgment is a call where the trade-offs cannot be solved by rule. It owns the outcome, not just the output.

THE SIX PERSONAS · 1 OF 2

Six Methods of Judgment

V Visionary

Decides what the company should pursue and what it should not. Reads weak signals — a customer asking for something nobody has built, a market shifting, a technology unlocking a door — and turns them into a clear bet. Names what success looks like before the work begins.

Strategic clarity — choosing what matters and explaining why so the rest of the company can align.

B Builder

Decides how the thing gets built. Architecture, technical trade-offs, the line between good-enough and not-yet. AI now writes the code, drafts the design, runs the tests. The Builder reads what AI produced and decides whether it meets the standard, where to push, and where to accept.

Technical judgment — knowing when AI's output is right, and when it has to be sent back.

G Guardian

Decides what risk is acceptable. Owns reliability, security, compliance, customer trust. The one who slows things down when they need to be slow, and signs off when they are ready. When something breaks at 2 AM, this is the owner. AI can flag the risks; the Guardian decides which ones the company will live with.

Risk judgment — knowing what could go wrong, and how much risk is worth taking.

D Dealmaker

Decides whether the product can be sold and at what price. Reads the buyer, the market, the competition, the timing. The one who closes hard deals and walks away from the wrong ones. AI can score leads and forecast pipeline; the Dealmaker decides what is worth pursuing and how to win it.

Commercial judgment — reading a buyer, pricing the product, closing the deal that serves both sides.

S Storyteller

Decides what story the company tells about what it has built. Positioning, narrative, brand, the language the market actually uses. AI can produce campaigns, copy variants, and performance data at volume; what it cannot do is decide which story makes the product matter to the people who should care about it. That call sits here.

Narrative judgment — turning what the company does into a story the market will recognise and remember.

A Advocate

Decides whether the customer will actually succeed. Onboarding, education, the moment a user gets stuck, the conversation that turns a complaint into a win. The voice of the customer inside the company before launch, not after. Reads the user before the user even knows what to ask for.

Customer empathy — seeing the product through the eyes of the person who has to live with it.

THE FULL PICTURE

Six Personas, One Band

Six personas, side by side. The table reads as a reference — but the question it raises is harder. Where do these judgments live in the company?

PERSONA	OWNS	KEY CAPABILITY
Visionary	Direction — what to pursue and why	Strategic clarity
Builder	Architecture — how to build it right	Technical judgment
Guardian	Reliability — safety, risk, production	Risk judgment
Dealmaker	Revenue — positioning, pricing, deals	Commercial judgment
Storyteller	Narrative — what story the company tells	Narrative judgment
Advocate	Adoption — customer success and trust	Customer empathy

These are not job titles. They are types of judgment. One person can hold several personas, and one persona can be held by several people — depending on the complexity of the work in front of them. What matters is that every one of these types of judgment has a clear owner — someone who makes the call and is accountable for the outcome.

> Departments were designed for execution. Personas are designed for judgment.

WHERE OWNERSHIP LIVES

The Band

Personas name the kinds of judgment a company needs. They do not yet say where those judgments live, or who is accountable for each one. A company can have all six personas present somewhere in the organisation and still have nobody clearly accountable for any single product line.

The container that solves this is the **Band**. A Band is the permanent home of a product line. It is the group of people who, together, hold every persona that product line needs — across all the initiatives that move through it over time. The Band stays. The work flowing through it changes. Every persona is named. Every persona is held by one or more people. Every person knows which personas they hold inside this Band.

A Band is the matrix of personas and humans. Every persona has a name on it. Every human knows which personas they hold.

A DEPARTMENT	A SQUAD	A BAND
Groups people by function. Owns a capability. No end-to-end accountability.	Cross-functional but narrow. Usually engineering + product. Owns features, not outcomes.	All six personas. Permanent. Owns a product line E2E. Accountable for the outcome.

> A Band is not a project team. It is the permanent home of a product line — with all the judgment it needs to succeed.

Most Bands serve external customers. Some — platform, infrastructure, security, design systems — serve internal customers: other Bands. The framework applies to both, with differences addressed after the Coherence Layer.

PART FOUR · CLOSING

How Ownership Comes Back

Once Bands are the unit, the conversation shifts. We stop asking *which department owns this*. We start asking *which Band owns this*, and *who in the Band holds which judgment*. Departments and professions remain — people still develop their craft inside them. But day-to-day work, decision rights, and accountability live in the Band.

Blurred ownership is what happens when AI gives everyone access to everything and no structure says who decides. The Band is the structure: a permanent group, named personas, named owners, one product line, ongoing accountability for the outcome.

This is the third kind of coherence the framework rebuilds: **ownership coherence** — every kind of judgment the work requires has a named owner, and every owner knows which judgments are theirs.

> The breadth stays. The accountability comes back.

PART FIVE

Designing Coherence

The Conductor, the rhythm,
and how to begin

PART FIVE · OPENING

The Conductor

The framework so far gives a company convergence points to rebuild direction, a book of record to rebuild alignment, and Bands to rebuild accountability. Each one solves a real problem. None of them asks the question that matters most: does it all fit together?

That question is the work of the **Coherence Layer**. And the Coherence Layer needs a leader — alongside the six personas who hold the domains.

That leader is the **Conductor**. The Conductor does not own a domain. The Conductor owns the connections between them — making sure what the Visionary chose, what the Builder is producing, what the Dealmaker is selling, what the Storyteller is positioning, and what the Advocate is preparing to deliver all add up to one coherent thing.

A Band without a Conductor has six strong players and no music. Even the finest musicians, playing at the same time, do not make a symphony — someone has to set the tempo, hold the beat, and listen across all the parts so the sounds add up to one piece.

> Six personas play. The Conductor listens.

CONDUCTOR RESPONSIBILITIES

Between the Convergence Points

Convergence points are decision moments. Between them, work is happening — personas building in parallel, AI generating outputs, signals appearing in every tool. The Conductor's daily job is to keep the book of record honest by curating what enters it. That work splits into two kinds of listening.

The data

Are all personas progressing against their commitments? Has the book of record changed since a stream last checked? Are the success criteria from the Selection Point still realistic given what we know now?

Tells you what already happened.

The room

Do people still believe in what they are building? Is someone working from an assumption that has quietly changed? Is there a concern nobody is surfacing?

Tells you what is about to happen.

The book of record stays current because an agent watches the personas' tools and proposes updates as signals appear — but the agent does not write to the record directly. Each proposal lands in the Conductor's queue: accept, reject, or amend. This keeps the record a coherent picture, not a feed.

The Conductor also answers the question AI makes harder than it sounds: *who knows what is affected?* When something changes, an agent can map the dependency graph mechanically. Deciding which dependents matter enough to act on now, and which people need to be told even when no artifact regenerates, is the Conductor's call.

> The dashboard shows the score. The Conductor reads the game.

THE CONDUCTOR · AN HONEST ASIDE

Equal among, different from

Six players, one Conductor — equal in voice, equal in standing.

But the Conductor is not equal in function. Each of the six personas is attached to a particular outcome — an attachment that makes them effective. It is also what makes them, individually, unable to see what they are not looking at.

The Conductor is the role that holds no attachment. The only person in the room who can ask the question no attached person will ask. That is not a deficiency. It is the source of the role's value.

The Conductor does not decide, does not overrule, does not block. When the team makes a decision that contradicts a rule the team set earlier, the Conductor's job is not to stop it — only to make sure the team sees the contradiction, names it, and decides knowingly. The team may go ahead. What is not allowed is going ahead by accident.

Equal in voice. Different in role. A peer in the room, a non-participant in the play. The instrument that turns implicit decisions into explicit ones.

> **The Conductor does not decide. The Conductor makes sure no decision is taken by accident.**

THE THREE LEVELS

Where Coherence Lives

The Conductor holds coherence inside one Band. But a company has more than one Band. So coherence has to be held at three levels — and each level has an owner.

Within the Band

Are all personas aligned on this product line?

Owned by the Conductor

Across Bands

Is the customer seeing one company?

Owned by each C-level leader

Company-wide

Does it all add up to one direction?

Owned by the CEO

C-level leaders sit above the Bands, not inside them. Each holds one professional layer across the whole company. Builders share one technical standard. Storytellers share one brand. Conductors share one operating rhythm. Which C-title carries which layer depends on the company. What matters is that each persona has an owner at the top.

The CEO holds the whole. Bands and layers, all adding up to one direction. That is the Coherence Layer.

When the customer is another Band

Most of this book describes Bands whose customer is the customer. But many companies also have Bands whose customer is *other Bands* — platform teams, infrastructure, identity, data platforms, security. Their work is real. Their gates are different.

Selection. A platform Band does not read customer signal. It reads what other Bands cannot read for themselves: architectural drift, duplicated effort, developer pain, the slow debt that every product Band pays for and none can fix alone. The Visionary reads the system, not the market. The signal that justifies the work is *other Bands' velocity will collapse if we do not do this.*

Launch. A product Band launches into its market. A platform Band launches into every Band that depends on it — at the same moment, regardless of where each dependent Band is in its own cycle. The same event hits a Band mid-Selection, a Band mid-build, a Band that just shipped yesterday. Launch becomes a coordination decision as much as a release decision.

Validation. A platform Band cannot be validated by adoption — its users are obligated, not optional. Validation asks a different question: *did downstream Bands move faster? Did duplicated work disappear? Did the drift we predicted get prevented?* The thresholds at Selection name these concretely so Validation has something honest to measure against.

Each of these raises the same harder question — when a platform Band's decisions affect four product Bands, whose convergence point governs?

CROSS-CUTTING CONCERNS · SPECIAL CASE (2 OF 2)

Whose convergence point governs?

Selection, Launch, and Validation all raise the same question in different forms: when a platform Band's decisions affect four product Bands, whose convergence point governs?

The platform Band's gates govern the platform decision — what to build, when it is ready to ship, when to deprecate. These are decisions the platform Band is closest to and most accountable for. No outside body should override them.

But there is a second question the platform Band cannot answer alone: *should this platform exist in this form at all, given where the company is going?* That question is not about the platform. It is about the architecture of the company — how the Bands fit together, what work belongs in shared infrastructure and what belongs in product Bands, where the company is heading and what substrate it needs to get there.

That question sits one level up. It belongs to the C-level layer — specifically to whichever C-level leader holds the relevant professional layer across all Bands. The CTO holds architectural coherence across all Builders. The CRO holds revenue coherence across all Dealmakers. When a platform Band's existence or its Launch timing is the question, the answer lives there.

Inside the platform Band: the same gates, with different signals. Above the platform Band: a coherence question that the gates alone cannot resolve. Both are needed.

> A Band whose customer is another Band is still a Band. The signals are different. The discipline is the same.

THE RHYTHM

How We Operate

The Conductor's work happens in three tempos.

Daily — the Conductor reads the room

Every day, the Conductor watches the book of record and listens to what is being said and what is not. Most days, nothing needs to move. Some days, something has shifted — a new signal, a quiet concern, a small drift between what was promised and what is happening. That is the day the routine earns its place.

Weekly — the Band meets

Once a week, the personas gather. Each one says what has changed in its part, what it has committed to next, and what it needs from the others. The book of record is updated against the conversation. It is short — under an hour. The point is not to report. The point is to listen across the parts.

At each convergence point — the company decides

At Selection, Shipment, and Validation, the Band brings the picture to the company. The call is made — start, ship, keep, or end. Commitments are renewed. The book of record moves to the next stage.

> Three tempos, one beat. The Conductor keeps it.

THE WHOLE FRAMEWORK, END TO END

Summary of the Logic

AI changes the cost of execution, the speed of execution, and the access to context. Three gifts. Three risks: **strategy divergence**, **fragmented execution**, **blurred ownership**.

Direction is held at three **convergence points** — the moments where the company chooses what starts, what ships, and what survives. Diversity is preserved between them. Direction is held at them.

Parallel work stays one through a **book of record** — one per initiative, continuously updated, with changes well managed, kept pruned to what matters now.

Ownership lives in the **Band** — the permanent home of a product line, where six domain personas (Visionary, Builder, Guardian, Dealmaker, Storyteller, Advocate) each hold a kind of judgment. Initiatives flow through the Band over time. The Band stays.

Coherence is held by the **Conductor** — the leader of the Band — and by the **Coherence Layer**: the rhythm of routines, commitments, and progress tracking that keeps every book of record honest and every parallel stream aligned.

> That is the whole framework, in one breath.

Three forces are gone. Three risks remain. The framework is not a defense — it is a redesign. The company that runs on convergence points, books of record, Bands, and the Conductor is not protecting itself from AI. It is what an operating model looks like when AI is part of how it works.

HONEST READING

When This Breaks

Every operating model has predictable ways of failing. Naming the failure modes early is how a team catches them while they are still small.

Conductor capture
The Conductor becomes the person who does the work instead of the person who holds it together. *The cure:* the Conductor reports what they are *not* doing each week, not what they are.

Book-of-record fatigue
The record becomes another document to maintain on top of everything else. *The cure:* AI drafts, humans approve. The moment the team is writing the record by hand, the discipline has already broken.

Convergence-point creep
Three convergence points slowly become five, then seven. Sub-checkpoints appear. *The cure:* any new checkpoint has to replace one, not add to it. Three is the ceiling, not the floor.

Validation Point skipped
The hardest convergence point to run is the one most likely to be quietly dropped. *The cure:* the Validation date sits on the company calendar, not the Band's. Missing it is visible.

Persona inflation
Six personas become eight, then ten. Every domain wants its own seat. *The cure:* hold the count at six. New domains map onto an existing persona, or they are not a judgment the Band needs to hold.

Two Bands, one dependency
Two product lines depend on the same shared platform. Both Bands claim accountability; neither has authority. *The cure:* a third Band owns the shared dependency as its product line.

THE FULL LOGIC

How It All Connects

WHAT HELD COMPANIES TOGETHER

Cost	Time	Siloed context
Gave direction	Gave alignment	Gave accountability

AI changes all three — cheaper, faster, full context ↓

WHAT THE SHIFT CREATES

Strategy divergence	Fragmented execution	Blurred ownership
Direction is no longer chosen	Streams stop seeing each other	Anyone can act, no one is accountable

Designed coherence replaces them ↓

WHAT REPLACES THEM

Convergence points	Book of record	The Band
Rebuild direction	Rebuilds alignment	Rebuilds accountability
Selection · Shipment · Validation	One source · One initiative	Six personas + one Conductor · One product line

The Coherence Layer
Connects the three into one operating discipline

Within Band	Across Bands	Company-wide
Conductor	Each C-level	CEO

WHO HOLDS THE JUDGMENT

V	B	G	D	S	A
Visionary	Builder	Guardian	Dealmaker	Storyteller	Advocate
Direction	Architecture	Reliability	Revenue	Narrative	Adoption

Conductor · Listens across all six. Holds coherence.

The main challenge of the AI era
is no longer execution efficiency,

---

but ***organizational coherence****.*

THE PRACTICE

The Practice

From theory to the first pilot —
a companion for the work ahead

FROM THEORY TO PRACTICE

The theory gave a structure. The Kit walks the first pilot.

The book offers a structure — a way of seeing the work, the personas, the convergence points, the drifts. A structure is useful, but it is not yet a practice. It does not describe what happens on Monday morning, in the meeting where this will either be taken seriously or quietly set aside.

To accompany the book, I built an implementation guide: the **Coherence Operating Kit**. It walks the Conductor through the first pilot — one Band, one Initiative, one quarter — and stays alongside the work the whole way.

The Kit carries a built-in tool — the Book of Record for the pilot, with the six personas and three convergence points already wired in — so the framework has somewhere to live from day one.

And it carries structured material for the Conductor: how to read the room, how to phrase the conversations that are hard to phrase, what to do when a persona pushes back, how to keep the record honest when it would be easier to let it drift. The questions the framework cannot answer on its own, the Kit answers — one situation at a time.

An example of what is inside follows on the next two pages.

> The book is the structure. The Kit is the companion that walks beside it.

What is inside the Kit

A readiness diagnostic

A short instrument the Band runs once, before Selection, that names which Band to begin the pilot in — and why that one and not another. Not every Band is equally ready. The diagnostic surfaces which one has the persona density, the operating room, and the conditions to hold the work.

A live Book of Record

The Book of Record for the pilot — six personas, three convergence points, every field already wired. The Band fills it in. The Kit keeps it coherent. Scope changes flow through approvals from the personas they affect. Decisions are logged in the room they were made in, with the forum named honestly. The record stays current because the framework keeps it current.

The first Selection Point, walked step by step

The conversation, the four documents that have to be in the room, the questions the Conductor has to ask aloud, and what to do when the room cannot yet answer them. The Kit holds the discipline of the Selection moment so the room does not improvise its way into a soft commitment.

What is inside the Kit — continued

The Signal Catalog

Thirty-five patterns the Coherence Layer watches for, organised across three tracks — operational signals from the tools the Band already uses, structural signals about how the framework itself is being installed, and coherence signals about whether the personas are still building the same thing.

Each pattern comes with a ready-made prompt for the source it watches — the issue tracker, the CRM, the security platform, the customer-success platform, the communication channels. Paste the prompt into the LLM of your choice, paste your tool's export, and the agent returns proposals in the Kit's schema. The Conductor triages: verify, monitor, or reject. Nothing enters the Book of Record by accident.

A few of the patterns: a ticket re-estimated past the threshold the Selection committed to. A CRM renewal note adding a condition the Selection never saw. A security finding with no Guardian owner. A chat thread where the Builder and the Storyteller are describing two different products.

The Conductor playbook

Ninety days of operating discipline, made specific. The daily reads, the weekly reads, what to do at each convergence point, the five drift patterns the framework deteriorates into if discipline lapses, twenty-five objections the Conductor will hear from the room, and the conversation scripts for the meetings that decide whether the pilot lives or dies.

A Validation kit

For the moment the pilot closes — what to keep, what to evolve, what to kill before scaling. The Validation Point is where the framework gets honest about itself. The kit holds that honesty.

CLOSING

Epilogue

Coherence, by design
and in practice.

EPILOGUE

What is coming will not be stopped. It is bigger than us. Some of it will be ugly. Some of it will be extraordinary. **If you cannot beat it, join it.** This book is how I am trying to join it — not adapting to AI, not surviving it, but operating inside it on purpose, with eyes open.

What I have written here is one piece of something larger I have been working on for a while. What used to hold organizations together by accident now has to hold by design. I chose software for this volume because it is the field I know best, and because it is where the change has already arrived. The principles travel further than software. But an idea has to survive a real test before I am willing to claim it for everything. This is that first test — the working hypothesis, with the SaaS-shaped evidence. The rest of the theory is in progress, written from inside the change it is trying to describe.

Thank you for reading this far. If a piece of this landed — or didn't — I would value hearing about it.

We are stepping into territory no one has mapped. The shape of work, the shape of organizations, the shape of what humans do all day — none of it is settled. We get to help decide. Very few generations have been handed that.

Ruth Amichay | ruthamichay.com

We are at the very beginning of a new era.

Don't miss the chance to shape it.

REFERENCE · 1 OF 2

Glossary

One-line definitions of the terms used in this book — both proprietary names and borrowed words used with specific meaning.

FORCES & RISKS

Strategy Divergence
When low cost makes everything look worth starting and direction is lost.

Fragmented Execution
When fast execution outpaces the alignment that used to come from natural pauses.

Blurred Ownership
When shared knowledge dissolves the silos that used to make ownership obvious.

DIRECTION (PART TWO)

Define · Produce · Operate
The three continuous stages every initiative passes through.

Selection Point
The decision point between Define and Produce. *Should this start?*

Shipment Point
The decision point between Produce and Operate. *Is this ready?*

Validation Point
A scheduled review of promise vs. outcome. Filters what lives and what dies after the first cycle.

STRUCTURE (PART THREE)

Book of record
The living document of one initiative — from idea to end of life, AI-maintained.

Initiative
A unit of work that goes through the convergence points as one — typically a feature, product line, or service.

Change Management
Part of operating the book of record well: distinguishing implementation changes from specification changes, tracking which streams each change affects, and making the implications visible.

Coherence Layer
The operating rhythm that keeps every book of record current and every parallel stream aligned.

Glossary — continued

ROLES (PART FOUR)

Persona
A method of judgment, not a job title. A kind of thinking the work requires.

Visionary
Defines what the company should pursue and why.

Builder
Owns how it gets made — architecture, technical strategy, build judgment.

Guardian
Owns reliability, security, compliance, and customer trust — and the call on what risk the company will live with.

Dealmaker
Owns the commercial side — pricing, positioning, deal strategy, market.

Storyteller
Decides what story the company tells about what it has built — narrative, positioning, brand judgment.

Advocate
Owns the customer side — adoption, success, voice-of-customer judgment.

Conductor
Holds coherence across personas. Asks the questions nobody else is asking. Runs the rhythm.

Persona density
The ratio of personas to humans inside a Band.

Band
The permanent home of a product line. A group of people who, together, hold the six personas and the Conductor required to drive that product line end-to-end.

BORROWED TERMS · SPECIFIC MEANING

Coherence
When every part of the company is building one thing — same product, same promise, same direction.

Direction
What the company chooses to point itself at. Controlled at the convergence points, not improvised.

Progress
How far an idea has travelled toward becoming value. Not how much has been built.

Simultaneous Execution
When work happens in parallel across all stages and functions, with no natural sequence.

Discipline
A practice that has to happen on purpose. Used here for the operating habits that keep the book of record alive.

Three forces held software companies together for decades. AI is weakening all three. This book is about what replaces them — an operating model where direction, alignment, and accountability are designed as one coherent system.

Ruth Amichay has spent 25 years building and running custom-made operations across software companies internationally. This is her framework for what comes next.

ruthamichay.com · claro.team